



LOG3000

Processus du génie logiciel

Hiver 2021

Présenté par :

Andi Podgorica (1955913)

27 février 2021

École Polytechnique de Montréal

Département de Génie Informatique et Logiciel

Code d'honneur

En répondant à cet examen à domicile ("take-home") de LOG3000,

je m'engage à :

- avoir complété cet examen par moi-même, sans l'aide d'autres personnes;
- ne remettre que mes propres travaux, qui ne contiennent aucun matériel plagié;
- ne divulguer pas les questions posées dans cet examen et les réponses aux questions à quiconque ne suit pas LOG3000 cette session;
- ne pas participer à des activités malhonnêtes visant à améliorer mes résultats ou à nuire aux résultats d'autres personnes. Signature : **Andi Podgorica**

Date et lieu : 27 février 2021, lieu Montréal

1 Question sur l'utilité des processus et de la gestion de projet (2 points)

a) Présentez deux utilités des processus de développement logiciel (0,5 point).

En bref, les processus servent à bien réfléchir sur la façon dont on va développer notre logiciel.

Premièrement, une des utilités des processus est de permettre une communication interne à l'équipe. Deuxièmement, une des utilités des processus est de permettre une évaluation des pratiques en place, c'est-à-dire de faire un diagnostic permettant de détecter des problèmes dans les manières de développer actuellement le logiciel.

b) Justifiez chaque utilité en présentant pour chacune un scénario concret où les processus de développement sont utiles. (1 point).

- Communication interne

Un exemple de situation concrète serait lorsqu'il y a des nouveaux arrivant dans une équipe de projet 3, c'est alors là qu'un processus défini peut les aider à bien comprendre à quoi sert leur travail ainsi que les informations et ressources qui leurs seront disponibles.

- Évaluation des pratiques en place par un diagnostic

Un exemple de situation concrète serait par exemple lors de la fin du premier sprint de projet 3 qui ne s'est pas bien déroulé en forme de rétroaction. En effet, on ferait une analyse après le sprint pour lier les défis vécus avec le processus de développement utilisé afin de l'améliorer dans le futur. En gros, on peut détecter des problèmes dans les manières dont nous développons le logiciel actuellement.

c) Décrivez les liens et les différences entre un processus et un projet. (0,5 point).

Différence: Tout d'abord, le processus permet de décrire comment l'information est transformée durant le développement du logiciel, tandis que le projet permet de planifier et d'assigner les ressources durant le développement logiciel. Aussi, le processus est un ensemble d'activités partiellement ordonnées ayant pour objectif la présentation de comment l'information est transformée, tandis que le projet est un ensemble de tâches ordonnées ayant comme objectif de planifier l'utilisation des ressources.

Lien: Le processus est partiellement ordonné d'activités qui sont reliées avec les tâches du projet. Dans le processus, les rôles (responsabilités et compétences) sont reliés avec les personnes physiques attirés du projet. Le processus a des itérations et boucles de rétroaction qui sont directement en lien avec le déroulement séquentiel du projet.

2 Question sur le choix d'un cycle de vie (4 points)

Pour les quatre contextes ci-dessous, expliquez (avec une justification) quel cycle de vie de base (incrémental, transformationnel, spiral ou cascade), serait le plus approprié :

a) Le développement d'un logiciel pour un système de contrôle ferroviaire qui contrôle la signalisation et les aiguillages ferroviaires. (1 point)

Spirale

Le cycle de vie spiral est approprié dans cette situation car il y a une analyse de risque très élaborée pour prioriser l'ordre des cycles. Étant donné qu'il faut s'assurer de faire un contrôle ferroviaire avec un faible niveau d'erreur, l'analyse de risque très élaborée du modèle spirale est appropriée et avec plusieurs étapes de vérification. De plus, nous ne tenons pas en compte les faiblesses du modèle spiral car la vitesse du développement n'est pas un requis important.

b) Le développement d'un jeu vidéo sur le thème de « l'amitié ». (1 point)

Transformationnel

Le cycle de vie transformationnel est le plus approprié dans cette situation car les exigences sont floues concernant l'amitié et il faut opter pour des cycles très rapides où on peut faire des corrections et recevoir des "feedbacks". Ainsi, grâce aux feedbacks, il sera plus facile d'évaluer le prototype en le montrant aux clients et découvrir le genre de jeu d'amitié qu'ils ont en tête. Par contre, un des inconvénients dans cette situation est que les clients ne connaissent pas le genre de jeu dont ils désirent et le prototype pourrait finir à la poubelle et le projet pourrait coûter plus cher.

c) Le développement d'une application Web qui synthétise des informations sur les cas de COVID19 dans les écoles primaires et secondaires. (1 point)

Cascade

Le cycle de vie cascade est approprié dans cette situation car nous connaissons assez bien les exigences et le risque qu'elles changent sont minimales. En outre, il y a peu d'aspects novateurs car l'application affiche les informations sur les cas de COVID19 et que ce sont seulement ces

informations qui changent, mais pas la forme du site. En gros, le code de l'interface ne va pas changer, mais les informations affichées sur le COVID qui changent. Ainsi, le cycle de vie cascade est le meilleur avec une faible incertitude technique et peu d'aspects innovateurs.

d) Le développement d'un système d'assistance pour la gestion des horaires qui fonctionne dans les téléphones intelligents, les tablettes, les ordinateurs et les téléviseurs intelligents. (1 point)

Incrémental

Le cycle de vie incrémental est approprié car il permet des itérations, des cycles rapides, de production du développement d'un système d'assistance se concentrant sur une fraction des exigences vu que le projet présent est fractionnable. De plus, les itérations sont indépendantes les unes des autres car il y a plusieurs catégories et différentes plateformes comme le développement mobile, tablette, ordinateur et téléviseur. Ainsi, ce modèle permet plus de flexibilité afin de faire des changements et une gestion du risque plus facile. Il est donc facile de séparer le développement en fractions pour les différentes plateformes. Une itération servirait au développement d'une plateforme à la fois. En effet, une itération permet de passer à travers les étapes de développement, les requis, le design, le code, les tests, et permet d'identifier les problèmes.

3 Question sur la qualité des exigences (4 points)

Partie A : Dans le contexte d'un service logiciel qui vise à prévenir les habitants du Québec par cellulaire, télévision et radio lors des calamités, indiquez pour chacune des exigences ci-dessous si l'exigence est fonctionnelle ou non-fonctionnelle.

Expliquez votre réponse.

a) Le système doit permettre aux responsables de la sécurité québécoise de créer des alertes adaptées à la situation. (0,5 point)

Fonctionnelle

C'est une exigence fonctionnelle car il y a une description des fonctionnalités du produit logiciel et que cela peut être construit en une tâche bien définie, celle de créer des alertes adaptées à

la situation. Ici, le but est de créer des alertes adaptées, mais la façon n'est aucunement spécifiée.

b) Le système doit envoyer des messages dans un format accessible aux personnes aveugles ou incapables de lire. (0,5 point)

Non-Fonctionnelle

C'est une exigence non-fonctionnelle car il y a une description des qualités du produit logiciel, celle de rendre accessible le format aux personnes aveugles. Il y a une description de la façon dont on veut permettre aux aveugles de recevoir les messages, et la façon est d'envoyer ces messages dans un format accessible.

c) Le système doit être capable d'envoyer des alertes en même temps aux plateformes différentes (cellulaire, télévision et radio). (0,5 point)

Fonctionnelle

C'est une exigence fonctionnelle car il y a une description des fonctionnalités du produit logiciel et que cela peut être construit en une tâche bien définie, celle de créer des alertes en même temps aux plateformes différentes. La façon de faire n'est pas spécifiée.

d) Le système doit enregistrer toutes les alertes générées et distribuées à des fins d'archivage. (0,5 point)

Non-Fonctionnelle

C'est une exigence non-fonctionnelle car elle décrit les qualités du produit logiciel et décrit de quelle façon on procède à l'archivage. La façon est celle d'enregistrer toutes les alertes générées et distribuées.

Partie B : Les exigences suivantes pour ce logiciel ont des problèmes.

D'abord, identifiez et expliquez les problèmes en fonction de chaque exigence. Ensuite, réécrivez chaque exigence pour corriger les problèmes.

a) Le système ne doit pas laisser personnel autorisé d'être incapable d'envoyer des alertes. (0.5 point)

- Qualité de la langue

La grammaire fait défaut, il y a des erreurs de français.

- Utilisation du négatif

Utilisation de négations, soit 'ne pas laisser personnel autorisé' et soit 'incapable' à éviter.

➤ Le système doit être capable de laisser le personnel autorisé d'envoyer des alertes.

b) Le système doit être très fiable. (0.5 point)

- Exigence doit être testable

Le mot 'très fiable' nous donne aucune indication claire sur la fiabilité du système.

➤ Le système doit être fonctionnel 98% du temps.

c) Le système doit être protégé contre des attaques à l'aide des technologies de sécurité de Java (progiciel java.security). (0.5 point)

- Indépendance du design

Ici, c'est trop spécifique sur comment le système est protégé, le logiciel de Java est spécifié.

➤ Le système doit être protégé contre des attaques à l'aide des technologies de sécurité.

d) Le système doit distribuer les alertes sur plusieurs plates-formes et les autorités doivent être en mesure de contrôler la distribution des alertes. (0.5 point)

- Exigences composées

Il y a deux exigences dans la même phrase.

- Utilisation du pluriel

Ambiguïtés lorsqu'on utilise le pluriel pour représenter les autorités.

- Le système doit distribuer les alertes sur plusieurs plates-formes.
- L'autorité doit être en mesure de contrôler la distribution des alertes.

4 Question sur la gestion du risque (5 points)

Vous êtes en train de faire votre projet intégrateur de 4^{ème} année. Ce projet consiste à démontrer la faisabilité (proof of concept) d'un outil logiciel pour une entreprise dans le domaine manufacturier. L'objectif de l'outil est de détecter automatiquement des défauts de fabrication en analysant la photo du produit sur la ligne de montage.

Il s'agit d'un projet relativement ambitieux pour une équipe d'étudiants avec peu d'expérience. Vous décidez donc de faire préalablement une analyse de risque afin de choisir l'approche de développement logiciel la plus appropriée pour la situation. Vous identifiez donc quatre risques, décrits dans le Tableau ci-dessous. Pour chacun des risques identifiés par vos collègues, proposez et expliquez une approche de gestion des risques.

ID	Risque	Probabilité	Impact	Niveau
a	Nous n'avons pas d'expérience dans le domaine de la reconnaissance d'image. Nous n'avons peut-être pas les compétences requises pour réussir le projet.	élevée	élevé	élevé
b	L'algorithme du réseau de neurones artificiels que vous avez prévu de passer six semaines à construire et à entraîner a de très mauvaises performances..	élevée	élevé	élevé
c	La librairie de reconnaissance d'image la plus populaire permet de reconnaître des objets simples dans des photos traditionnelles. Elle n'est peut-être pas appropriée pour la reconnaissance de défauts de fabrication dans un cadre industriel.	Moyenne	Moyen	Moyen
d	Le code finale est mal structuré, mal documenté et difficile à comprendre. Sa maintenabilité future est faible.	Moyenne	Faible	Faible

- a) Une approche de gestion de risque proactive, celle d'éviter, car on a le temps d'apprendre sur le sujet afin d'éliminer le risque qu'on n'ait pas assez de connaissances. De plus, dans ce contexte le risque est mentionné avant le commencement du projet. Nous avons le temps de nous réorganiser et d'apprendre sur la reconnaissance d'image pour éliminer le risque de ne pas avoir les compétences requises pour réussir le projet.

- b) Une approche de gestion réactive, celle de contourner, car nous pouvons choisir d'implémenter un nouvel algorithme vu qu'on se rend compte que l'algorithme présent a de mauvaises performances. Ainsi, nous ne continuons pas à travailler avec un mauvais algorithme pouvant nous compliquer la tâche dans les futures semaines en essayant de le corriger et de le faire fonctionner.

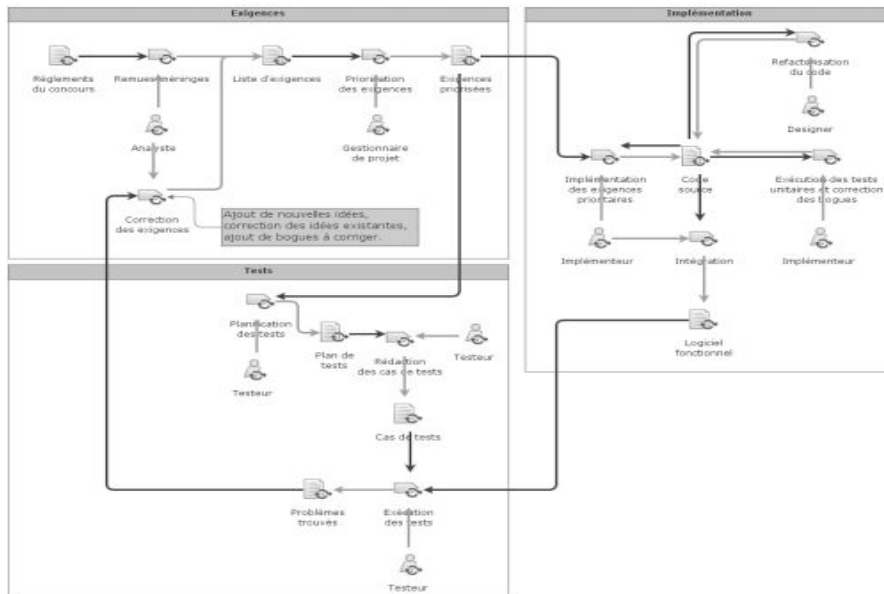
- c) Une approche de gestion proactive, celle de mitiger, car nous ne sommes pas sûr si la librairie va fonctionner, ainsi nous voulons faire des itérations plus courtes afin de corriger le tir rapidement si la librairie de reconnaissance d'images n'est pas appropriée. Alors, l'approche de mitiger nous permet de réduire l'impact et de ne pas perdre notre temps si on se rend compte que cette librairie ne fonctionne pas lors de nos courtes itérations.

- d) Une approche de gestion "sans-action", celle d'accepter, car le niveau de risque est faible. Cette approche permet de vivre avec certaines erreurs, mais de les surveiller. Il est peu probable, car niveau faible, que le code final soit mal structuré, mal documenté et difficile à comprendre alors l'approche sans-action est la bonne dans cette situation.

5 Question sur l'analyse qualitative de processus (4 points)

Vos collègues participent au concours « Montréal virtuel », dans lequel les groupes participants sont invités à développer une application de réalité augmentée / virtuelle qui expose Montréal aux touristes virtuels. Les règles (c.-à-d. la vision) sont très générales

afin de permettre aux participants de se démarquer avec des idées innovantes. Vos collègues ont planifié une approche incrémentale classique pour la réalisation de la démo du concours. Leur processus planifié est illustré dans la Figure ci-dessous.



Comme tout processus, le choix de vos collègues a des forces et des faiblesses.

a) Décrivez au moins trois (3) problèmes auxquels vos collègues sont exposés en suivant ce processus (3 points).

Il y a des éléments de trop qui ne sont pas très pertinents dans le cadre de cette démo, comme la rédaction des cas de tests et du plan de tests. En effet, rédiger des cas de tests n'est pas très nécessaire pour une démo et peut résulter à un manque de temps. De plus, l'activité de refactorisation du code par le designer n'est pas pertinente car c'est une démo et la démo n'a pas besoin d'avoir du code maintenable. Aussi, en utilisant le modèle incrémental, des problèmes d'architecture peuvent survenir car les requis ne sont pas connus au départ et risquent d'être modifiés. Il y a trop de focus sur la qualité du produit tandis que le produit peut évoluer car c'est une démo. Aussi, il n'y a pas de communication entre les testeurs et les implémentateurs, ce qui peut résulter à des bogues qui ne sont pas détectés. En outre, l'activité remue-méninges permet d'arriver à une liste d'exigences importantes. Le processus ci-dessus suit un cycle de vie incrémental SCRUM qui marche bien si on a principalement une quantité d'exigences importante pour former un "Product Backlog". Dans le cas du concours, ceci est risqué car les besoins initiaux ne sont vraiment pas clairs. Le cycle de vie incrémental a des

inconvenients pour ce projet car dans le modèle incrémental il est difficile de déterminer la fin du projet, tandis que pour la démo il faut une date de fin.

b) Décrivez, textuellement et/ou à travers un modèle, une correction au processus qui permettrait de mieux gérer un des problèmes identifiés précédemment. Vous n'avez à proposer qu'une correction. Vous devez cependant justifier votre correction (1 point)

Correction proposée: Une correction au processus afin de mieux gérer les problèmes identifiés est de ne pas faire de plan de tests car le plan de tests c'est pour plus tard afin de trouver des bogues et les cas de tests c'est pas réalisable pour simplement une démo. On a pas besoin d'un plan de tests pour une démo, car la démo n'est pas un projet de long terme et définitive.

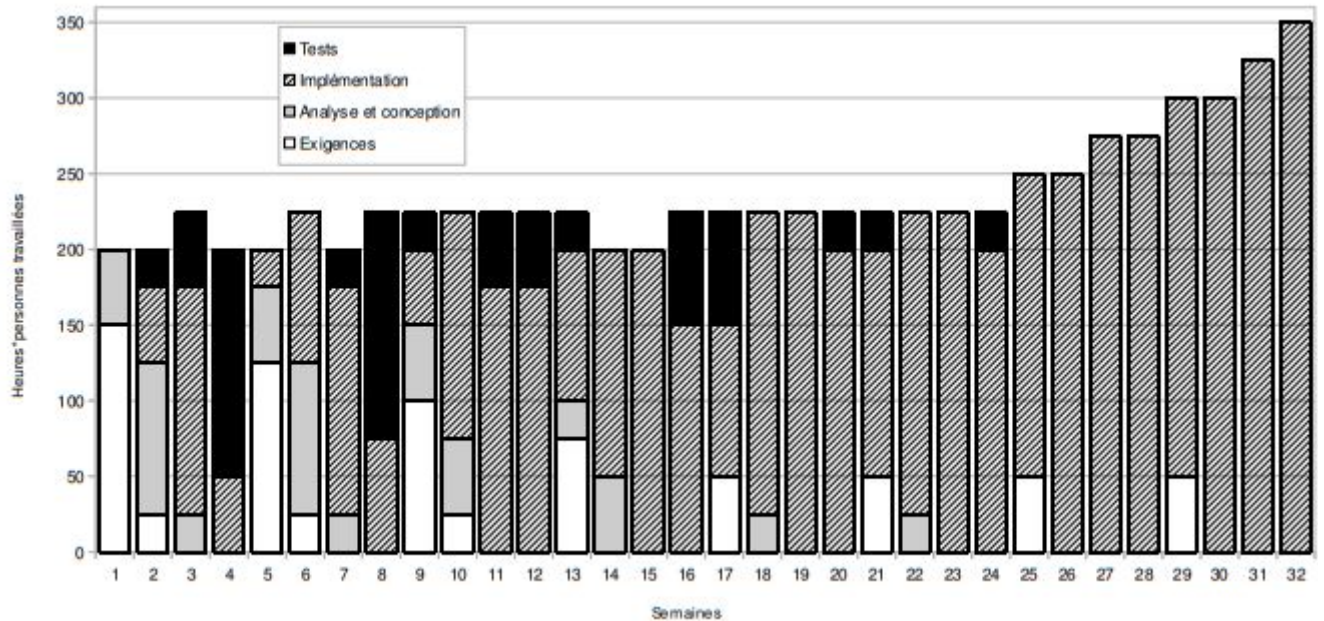
6 Question sur l'analyse quantitative de processus (4 points)

Vous êtes gestionnaire d'une équipe de maintenance pour une grande organisation. D'autres équipes sont responsables de développer les logiciels.

L'une des équipes de développement vient tout juste de livrer un nouveau logiciel, et vous vous retrouvez maintenant responsable de la maintenance. Cependant, vous avez entendu parler des difficultés rencontrées lors du développement de ce logiciel. Cette information vous laisse soupçonner que ce sera un défi pour votre équipe.

Vous avez réussi à obtenir les données de déroulement du développement logiciel. Ce logiciel a été développé en 32 semaines par une équipe de six personnes qui travaillaient à temps plein. Cela représente des horaires de 37,5 heures par personne par semaine, ou bien 225 heures par semaine pour l'ensemble de l'équipe.

La figure ci-dessous présente le détail de l'effort passé sur le projet, selon les quatre disciplines principales du développement logiciel.



a) En utilisant les données de la figure, identifiez et décrivez le cycle de vie utilisé pour le développement de ce nouveau logiciel (1 point).

Dans cette figure, le cycle de vie est incrémental car il y a une itération aux quatre semaines. Nous savons que le modèle incrémental a des itérations de 2 à 6 semaines, ce qui correspond à celui du cycle de vie de ce projet, qui est de 4 semaines. De plus, nous pouvons constater un travail sur les exigences à toutes les quatre semaines s'ensuivant par un travail sur le design, pour les semaines dont ils les font. Ensuite, on peut constater un travail en implémentation suivi de tests, pour les semaines dont ils les font également. Bref, on peut conclure que ceci correspond à des itérations de style cascade correspondant donc au modèle incrémental.

b) En utilisant le cycle de vie et les données de la figure, décrivez des problèmes potentiels concernant la maintenance (1.5 point).

Un des problèmes qu'on peut constater est que peu de tests sont effectués, principalement vers la fin du projet. On peut également constater que les dernières itérations ont été des sessions chargées de travail sur le code car le travail sur le design n'est plus effectué. Vers la fin du projet nous remarquons qu'il y a des nouvelles exigences qui sont demandées, tandis qu'il n'y a pas de phase de conception et de design, ce qui fait en sorte que les employés ont codé sans avoir fait de design. Cela a bien entendu amené à des difficultés de maintenance de code.

Donc, par le manque de ce travail, on est portés à croire qu'il y a peut-être des problèmes au niveau du design, mais également qu'il y a probablement beaucoup de bogues qui n'ont pas été détectés, ce qui concerne la maintenance.

c) En tant que gestionnaire de l'équipe de maintenance, et tenant compte des problèmes décrits précédemment, que feriez-vous afin de mitiger les défis de maintenance liés à ce nouveau logiciel (1.5 point) ?

En tant que gestionnaire de l'équipe de maintenance, afin de mitiger les défis de maintenance liés à ce nouveau logiciel, en sachant qu'il y a des bogues causés par l'absence de tests, je commencerais par faire une évaluation de la qualité architecturale et par la suite je ferais du "refactoring" de code. De plus, je ferais des tests unitaires et d'intégration testant la nouvelle architecture afin de déterminer les bogues introduits. Pour terminer, je ferais la correction des bogues ainsi que l'implémentation des tests pour les fonctionnalités du nouveau logiciel afin de s'assurer de la qualité de ce dernier.

7 Question sur la modélisation de processus (6 points)

Il est possible d'installer dans les appareils mobiles Android des systèmes d'exploitation différents de ceux installés par le fabricant. Ces systèmes d'exploitation, aussi appelés ROM, sont devenus populaires pour différentes raisons. Par exemple :

— CyanogenMod/LineageOS permet une meilleure configuration de l'appareil mobile, tout en supportant les dernières nouveautés d'Android;

— ParanoidAndroid offre une plateforme beaucoup plus sécuritaire que le système d'exploitation Android de base;

— SlimRoms est un système d'exploitation très léger assurant une performance adéquate même dans de vieux appareils mobiles.

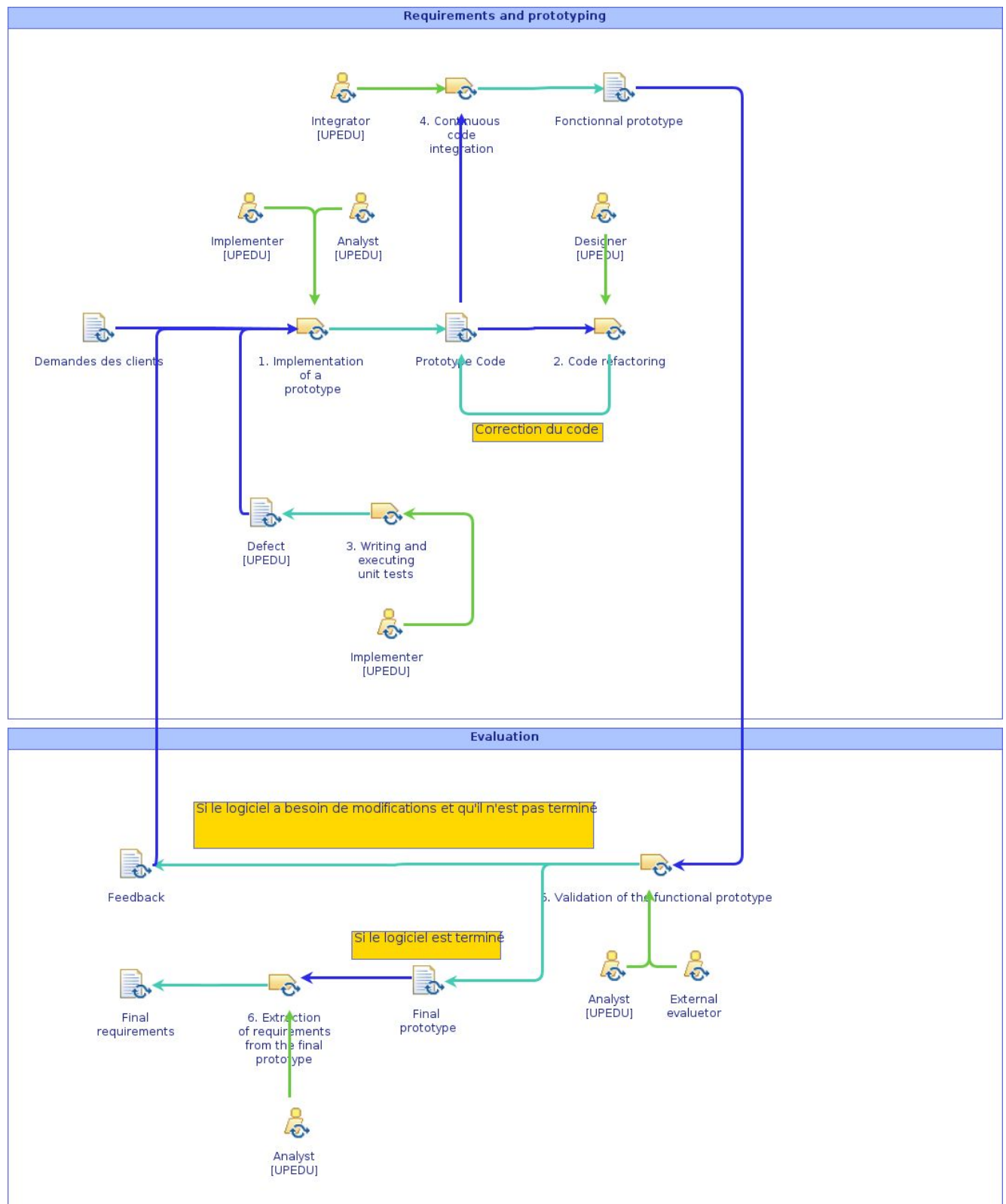
Vous et trois de vos amis décidez donc de monter votre propre système d'exploitation pour les appareils mobiles utilisant Android. Votre objectif est de produire un système d'exploitation accessible et facile à utiliser pour les personnes âgées qui voudraient avoir un premier téléphone intelligent.

Vous travaillez cependant sur ce projet à temps perdu, surtout la fin de semaine. Ainsi, vous voulez vous assurer de bien utiliser ce temps. Vous décidez donc de planifier le travail grâce à un processus.

a) Choisissez le cycle de vie qui vous semble le plus approprié dans ce contexte. Justifiez votre choix (1 point).

Le cycle de vie transformationnel est approprié dans ce contexte car on veut bien utiliser le peu de temps que nous avons et permet de faire des itérations très rapides allant d'une journée à deux semaines, ce qui rentre bien dans le cadre d'une fin de semaine. De plus, dans le modèle transformationnel, il est possible de faire un prototypage pour faire la conception centrée et validée par les personnes âgées, car c'est pas des personnes habitués avec la technologie et le prototype pourrait rapidement changer selon leurs besoins.

b) Modéliser un processus de développement complet pour ce projet basé sur le cycle de vie identifié précédemment (5 points).



8 Réflexion (1 point)

Réfléchissez aux sujets du cours que nous avons couvert jusqu'à présent. Énumérez les sujets qui, selon vous, sont les plus utiles (0,5 point) et les sujets dont vous souhaitez en savoir plus, mais qui manquent dans le cours (0,5 point).

Sujets les plus utiles:

- Discipline des tests
- Discipline des requis
- Discipline d'implémentation
- Discipline d'analyse et conception

Sujets manquants:

- Modèles de cycle de vie linéaires
- Contrôle des processus (durée du processus, coût du processus, production effective du processus)
- Cartographie des processus de haut niveau (donne un aperçu des processus et objectifs d'une organisation, précision des départements et rôles y participant au projet)
- Processus de changement transformationnel